LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-CONF-236751

NARAC SOFTWARE QUALITY ASSURANCE: ADAPTING FORMALISM TO MEET VARYING NEEDS

Hoyt Walker, John S. Nasstrom, Steven G Homann

November 26, 2007

American Nuclear Society 2nd Joint Emergency Preparedness & Response and Robotics & Remote Systems Topical Meeting
Albuquerque, NM, United States
March 9, 2008 through March 12, 2008

Disclaimer

UCRL-CONF-236751
Submitted to American Nuclear Society 2nd Joint Emergency Preparedness and Response & Robotics and Remote Systems Topical Meeting
March 9 - 12, 2008, Albuquerque, New Mexico

NARAC SOFTWARE QUALITY ASSURANCE: ADAPTING FORMALISM TO MEET VARYING NEEDS

Hoyt Walker, John Nasstrom, Steve Homann

National Atmospheric Release Advisory Center
Lawrence Livermore National Laboratory
7000 East Avenue, Livermore, CA, 94551-9234
hwalker@llnl.gov

The National Atmospheric Release Advisory Center (NARAC) provides tools and services that predict and map the spread of hazardous material accidentally or intentionally released into the atmosphere. NARAC is a full function system that can meet a wide range of needs with a particular focus on emergency response. The NARAC system relies on computer software in the form of models of the atmosphere and related physical processes supported by a framework for data acquisition and management, user interface, visualization, communications and security. All aspects of the program's operations and research efforts are predicated to varying degrees on the reliable and correct performance of this software. Consequently, software quality assurance (SQA) is an essential component of the NARAC program.

The NARAC models and system span different levels of sophistication, fidelity and complexity. These different levels require related but different approaches to SQA. To illustrate this, two different levels of software complexity are considered in this paper. As a relatively simple example, the SQA procedures that are being used for HotSpot, a straight-line Gaussian model focused on radiological releases, are described. At the other extreme, the SQA issues that must be considered and balanced for the more complex NARAC system are reviewed.

I. INTRODUCTION

The National Atmospheric Release Advisory Center (NARAC) provides tools and services that predict and map the spread of hazardous material accidentally or intentionally released into the atmosphere. Located at the Lawrence Livermore National Laboratory, NARAC is a national support and resource center for planning, preparedness, real-time emergency response, and detailed assessments of threats and/or incidents involving a wide variety of hazards, including nuclear, radiological, chemical, biological or natural emissions. NARAC products provide information on affected areas, potential casualties, health effects, and protective action guides to assist decision makers.[1,2]

NARAC is a distributed system, providing modeling and geographical information tools for deployment to an end user's computer system, as well as real-time access to global meteorological and geographical databases. The core of the capability is a suite of models ranging from simple, fast running Gaussian models to advanced three-dimensional model predictions run at the national center. These models and their supporting systems must be extensively verified and validated in order to insure that they have been implemented properly, produce realistic predictions, and are reliable in emergency conditions. NARAC's Software Quality Assurance (SQA) procedures are a component of this on-going verification and validation effort. As a consequence, the details of SQA in NARAC are evaluated, adapted and improved with the goal of improving NARAC's overall Quality Assurance within funding and other resource constraints.

To illustrate some of the considerations involved in defining NARAC SQA procedures, two perspectives on the system will be taken. First, the procedures being used for a relatively simple but very important component of the system, HotSpot, which also functions as a stand-alone modeling capability, will be described. Then, the issues associated with the overall system will be presented, with the goal of illustrating the need for adaptability in dealing with SQA for a system such as NARAC. Some additional background on these two systems follows.

I.A. HotSpot

One important component of the NARAC system is the HotSpot model, which is a Gaussian model that provides specific support for radiological problems. It is used to provide rapid emergency response and for safety analyses. For safety analysis applications, this model is

being placed in the DOE Safety Software Toolbox[3], contingent on completion of certain tasks. These tasks include the addition of a capability to utilize historical meteorological data and completion of documentation to meet DOE standards. Also, the incorporation of requirements tracking, change and test tracking, and configuration management is required. These last issues are being met by integrating parts of the HotSpot development process with the more formal NARAC system SQA procedures. This integration is tuned to meet the stringent requirements for Toolbox codes without placing an undue burden on HotSpot development. HotSpot has been developed and is maintained and extended by a single developer, who is also the scientist. As a consequence, some SQA procedures can be simpler and more focused. On the other hand, some HotSpot SQA procedures are being integrated with NARAC procedures to improve the HotSpot procedures and to broaden the knowledge of HotSpot development to others in NARAC.

I.B. NARAC System

The goal of the NARAC system is to make available to emergency responders and others the highest quality model predictions possible given the maturity of atmospheric models, the available data and technology, and within constraints of computational speed and budget. Thus, the quality of NARAC model predications is a function of many issues including SQA procedures. Constant evaluation of the system performance in real responses and exercises demonstrates that the weaknesses in the ability to produce accurate results are often the result of inadequate data or poorly represented physical processes rather than software defects. Consequently, while strong procedures for SQA are critical, resources must also be spent on improving the models, acquiring better data and taking effective advantage of technology. Maintaining the necessary balance, in the context of budgetary constraints, is an on-going challenge for the program.

In contrast to HotSpot, which is a single model and support system focused on a relatively narrow area of hazard assessment and emergency response, NARAC is a suite of models that range from simpler models (including HotSpot), through sophisticated 3-dimensional models appropriate for emergency response, to computational fluid dynamics (CFD) models, which predict complex flow around 3-dimensional structures such as buildings. NARAC responds to all hazards and at all physical scales from the building scale to the hemispheric scale. The NARAC concept of operations extends beyond the provision of a single, automated estimate of a dispersion pattern, by providing a service to emergency responders that is iteratively tuned in real-time to meet evolving needs, to take advantage of improved data describing the situation, and to actively incorporate all validated measurement data into the model predictions.

The broad and challenging mission of NARAC, which includes the goal predicting the real world dispersal pattern in all situations, requires that SQA efforts be evaluated within the context of this mission, not as an isolated technical issue. In fact, essentially all activity within the NARAC/IMAAC program can be considered as complementary parts of a general quality assurance focus. Section II introduces this perspective on the program and presents the SQA problem in this context. This lays the groundwork for the discussion in Section III that itemizes the key elements of SQA and provides more detail on how SQA processes are used in NARAC and integrated into the overall program.

II. QUALITY ASSURANCE IN NARAC

The NARAC program can be viewed as having three major building blocks: 1) operations, 2) model research and development and 3) hardware and software systems. These parts interact constantly to meet mission requirements. This interaction is facilitated by physical proximity with almost all of the staff located on the same floor of a single building. These sections can be represented as in Figure 1.

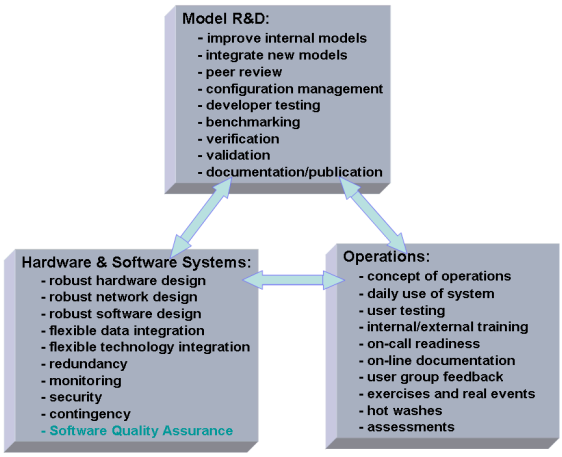

Figure 1. NARAC Quality Assurance Components

Each building block of the system plays critical roles in quality assurance. The Operations team is responsible for emergency response and interactions with external customers. Their outlook tends to be tactical and is focused on producing the best possible results given the state of the system, the available information and the time constraints that exist during a response. The Model Research and Development (R&D) team works towards improving the quality of the NARAC service by

improving the underlying science and modeling available in the operational model suite. Their outlook tends to longer term and involves interactions with the scientific community outside of NARAC to identify the most appropriate science to add to the system. The Hardware and Software Systems team provides the framework that allows operations to function effectively and supports the inclusion of new and improved capabilities provided by the Model R&D team. The Systems team looks at both the short term needs of Operations and the longer term requirements implied by the work of the Model R&D team while evaluating the available technology and data sources to determine how to most effectively meet the needs of the overall system. A few specific comments about each area will clarify the roles of each team in improving the quality of the NARAC service.

II.A. Model Research and Development

Given the mission of providing the best possible predictions to emergency responders within existing time and cost constraints, NARAC relies on constantly improving its core models, both in term of the science the models are based on and the details of their implementation, with the goal of increasing the fidelity that can be brought to bear on hazardous releases in emergency response time frames. In addition, a large number of collaborations are on-going to integrate improved science and modeling technology developed elsewhere both into the core models and integrating new component models that represent physical processes that have been previously ignored but have real and important impacts on dispersion patterns. These new and improved model components are involved in an on-going verification and validation effort that includes three key components: 1) validation against analytic tests, 2) validation against a constantly growing suite of tracer experiments and 3) validation against the data from real responses. The efforts of the model developers include many key SQA activities such as developer testing and repeatable benchmark test suites, supplemented by peer review and publication in the scientific literature.

II.B. Operations

The Operations team is an important element in NARAC quality assurance and, in fact, the concept-of-operations for NARAC puts the operational staff at the core of the NARAC service. This service extends far beyond the automated running of a dispersion model, regardless of how sophisticated. Recognizing the real incidents are more complex and unique, NARAC always supplements a fast, automated model execution, with at least one quality controlled re-execution of the model suite where all inputs have be reviewed by the expert staff at NARAC, all model outputs have been evaluated and, in many cases, model inputs adjusted, to produce a product that reflects a greater degree of confidence. Generally, the response to significant releases includes measurements of air or ground contamination. Subsequent NARAC model executions incorporate these measurements to improve the predictions as well as incorporation custom elements tailored to meet the needs of the emergency responders. Thus, a NARAC response to a real event will always involve a few and often more than a dozen model executions that iteratively converge on an accurate picture of the release and reflect the best available data and the expertise of the NARAC staff.

The operations team performs many tasks in addition to emergency response such as scientific studies in collaboration with the Model Development team and assessments for external customers and well as formal testing of the system. It is also important to note that most of the non-response activities of Operations are performed using the same software that is used in operations so that essentially all activities are constantly verifying and validating the performance of the system.

II.C. Hardware and Software Systems

The systems development team has an on-going mission to not only meet the current needs of Operations by developing and maintaining reliable, correct systems but also by integrating new data sources (e.g., geographical and meteorological data) and technologies that improve the fidelity and performance of the system. Given the rate of growth in potentially applicable data sources and explosion of potentially applicable technology, this is a challenging task. In meeting the needs of Operations, the Systems team is responsible for development and maintenance of redundant data sources and pathways, so that the system can perform acceptably even when data paths are broken (often due to events outside the NARAC system such as external network or remote installation problems). Given the complexity of the system and its reliance on external systems, the systems team also provides many levels of automated monitoring from basic disk and network use diagnostics to automated external problem submissions that test the availability and readiness of the system and inform staff via e-mail and pages whenever critical resources are out of compliance. Last, but not least, the systems team focuses on all aspects of software quality assurance to provide Operations and the Model Development teams with a reliable, correct environment to complete their tasks. Details of the SQA processes will be described in Section III.

II.D. Quality Assurance Summary

From these few comments, it is reasonable to say that essentially all activity within NARAC is focused on ensuring the quality of current products and supporting the improvement of these products in the future. The balance between efforts to avoid and detect software defects as part of SQA must be balanced against the need to incorporate new capabilities, data and technology in a cost-effective manner. This balance shifts as monitoring and customer interactions identify weaknesses in the system, whether in the software implementation or missing modeling processes or ineffective presentation to clients. Constant evaluation of all aspects of the system allows NARAC to adjust the balance and to keep it in a near optimum state given the availability of funding and other resources.

III. NARAC SQA ACTIVITIES

Ten activities are identified in the DOE documents on SQA.[3,4] These activities span the issues that need to be considered in defining the procedures that are used to develop and maintain software in systems such as NARAC and HotSpot. Each of these topics is discussed below with comments about both HotSpot and NARAC.

III.A. Software project management & quality planning

III.A.1. HotSpot

HotSpot project management has been based on the personal records of the developer and this has been adequate in the past. However, the incorporation of HotSpot as a component of the NARAC system implies an institutional support beyond that associated with a single developer. Consequently, version control, requirements tracking and problem reporting are being integrated into NARAC operational procedures (see comments on these issues below) and task prioritization will involve a wider audience including NARAC Management and the three teams described above.

III.A.2. NARAC System

Project management in NARAC is a customized mixture of tools and procedures that have grown over time to represent an appropriate balance between formalism and flexibility. The tools involved include Gantt charts, responsibility matrices, Java Café, EXCEL spreadsheets, Word and text documents. The details of use evolve over time with more effective approaches being continued and refined and less effective approaches being eliminated. The central tool in NARAC project management is Bugzilla, which was developed and is marketed as a defect tracking tool. Bugzilla is customizable and NARAC has extended this tool to handle requirements tracking and testing. It is used for all developer tasking including major efforts and minor enhancements. Large tasks can be broken down and conveniently managed as subtasks. By using a single tool for most of the key project management tasks, the overhead of implementing these practices is minimized. While a balance between formalism and flexibility is always important, the general trend has been towards increasing formalism.

Long-term planning is a negotiation between NARAC management and the development teams. Significant changes have been implemented in an evolutionary manner, however, the scope of some technology changes require special consideration. NARAC is currently evaluating proposals for significant technology upgrades aiming towards the most effective 3-5 year vision direction for the system.

III.B. Risk management

III.B.1. HotSpot

HotSpot, as a single developer project, has the associated efficiencies of minimal intra-project communication requirements and avoids potential confusion that can occur between team members with different backgrounds and perspectives. However, it can suffer from increased risk due to lack of independent evaluation and testing, lack of flexibility in responding to competing requirements because only one person can respond, and reliance on personal record keeping and memory for documenting and tracking requirements. Recently, HotSpot evaluation and testing has involved a NARAC model developer as part of the integration of the model into the NARAC system. This independent testing will be increasingly formalized to mitigate this aspect of risk. The risk of a single developer is largely accepted as necessary at current funding levels; however, the code is being added to the NARAC version control system, which is a first step towards reducing this risk. Integrating HotSpot requirements and problem tracking with the Bugzilla system will mitigate risk in managing requirements.

III.B.2. NARAC System

Risk management in general, is an on-going activity in NARAC. This includes working to reduce or eliminate weaknesses identified in responses, exercises, testing and to minimize strategic risks identified by management on

the basis of sponsor interactions. In this atmosphere, consideration of software risks fits naturally and there are on-going discussions on how to reduce these risks. In minimizing and managing risks, NARAC is helped by the presence of an experienced and dedicated staff. Table 1 illustrates this:

Experience	0-5 years	5-10 years	10-20 years	20-35 years
Total	1	2	5	4
NARAC	3	2	4	3

Table 1. Number of NARAC software development staff with various years of total and NARAC-specific experience.

Numerous other factors mitigate risk. These include the co-location of all staff, which facilitates close interactions and responsiveness. The daily use of operational tools in non-operational work can identify defects or other weaknesses in the system, typically long before an offending component is used in an operational setting. The monitoring tools and procedures mentioned above identify problems as they arise, which speeds the diagnosis and correction of the problem. The design and implementation skills of the development staff are constantly improving as are the tools used to facilitate development (e.g., compilers, debuggers, code analyzers).

Factors that increase risk include the growing range of modeling, data and customers requirements, which implies a general growth in system complexity. This is mitigated by improved design and software tools. Also, the advantage of an experienced staff implies the risk of the loss of key personnel to retirement and health issues. The difficulty of transferring knowledge to younger staff is compounded by the presence of both growing requirements and increasing pressure to control costs.

III.C. Configuration Management

III.C.1. HotSpot

Configuration management for HotSpot has been undocumented but strictly controlled as a result of the developer's procedures. However, the integration of HotSpot into the NARAC system places an increased value on more documented and standardized procedures. Consequently, HotSpot updates will be coordinated with the NARAC version control system. In particular, the HotSpot sources associated with each release, as well as a number of past releases, will be entered into the version control system. Further, integration of HotSpot procedures will occur as experience is gained in this area.

III.C.2. NARAC System

All components of the system are in version control, including utilities, testware, and some test data. Note that some test data is very large (e.g., geographic and meteorological data) and is managed as an independent resource. An important configuration management issue is the clear separation between development and production environments. This avoids the risk of polluting a production environment with corrupt data or software during development and testing. Highly formal and documented procedures are followed during production updates to ensure that the new software is appropriately tested and that one or more production systems are always available. The on-going system, network and software monitoring tools help in this process. The system performance, procedures and detailed documentation in this area are evaluated regularly and improved as needed. Note that changes to the configuration of the system in association with new versions of the software are entered into and managed using the Bugzilla system.

III.D. Procurement and Implementation

III.D.1. HotSpot

HotSpot is built using the Microsoft Visual Basic V6.0 and Macromedia RoboHelp X5, which are well accepted tools and are developed using appropriate SQA procedures.

III.D.2. NARAC System

NARAC tracks procurements of system, packages, tools and data using EXCEL spreadsheets supplemented by text documents. The core tools used within the operational system are selected and maintained based on:

- Support for require functionality
- Ease of integration
- Vendor reputation and previous experience
- Deployment constraints
- Cost

Such tools are continuously evaluated in the light of evolving technology and changing system requirements. These tools are replaced as better options become available and as development resources are available. The maintenance for these tools is selected at a level appropriate to the potential impact of failure.

A variety of tools are used by various staff for research and assessments that are not used in normal operations (e.g., some visualization and statistical analysis

packages). These are not evaluated as closely for SQA issues.

III.E. Requirements Management

III.E.1. HotSpot

As discussed previously, HotSpot management of HotSpot requirements is being integrated with NARAC processes including the use of Bugzilla. In addition, the discussion and prioritization of requirements involves a collection of staff from the NARAC program in addition to the single developer.

III.E.2. NARAC System

NARAC software requirements come from two main sources. Changes are proposed by operations and other users using Bugzilla. The changes include defects that are discovered during use of the system but are more commonly incremental enhancements that address missing features or tools that allow users to function more efficiently or produce better products. Operations also deals directly the external clients of the system and their requests incorporate issues identified by these clients. Note that all users of the system can submit change requests, including developers. This is an advantage because tools are added by developers to meet needs for new work often exercise the system in novel ways. By enhancing the rest of the system as new capabilities are developed, fewer problems occur when the new capability is introduced to operations. Such changes are largely tactical in nature. In contrast, strategic changes are driven by requirements set by sponsors and managers on the basis of a variety of larger-scale issues associated with meeting the NARAC mission in the eyes of the sponsors of the program. These changes typically require longer term efforts involving staff from all three teams and, consequently, must be matched with current and future funding realities.

Development priorities are set in a negotiation between the development staff, sponsors, operations, model developers and managers that combines the requirements with estimates of development costs and external constraints such as the need for a feature before a specific scheduled exercise such as the TOPOFF national-level exercise series. The tracking of the requirements is managed using Bugzilla. Generally, all requirements being actively worked by development staff are maintained in the extended Bugzilla system.

III.F. Design and Implementation

III.F.1. HotSpot

The HotSpot code is largely in maintenance mode. The HotSpot design is documented using RFF Electronics RFFlow V5 using context diagrams and flow charts. A migration of Microsoft Visio is being considered to provide more flexibility. The Visual Basic development environment is utilized heavily, supplemented by Macromedia RoboHelp X5 to implement an embedded help feature. The HotSpot GUI performs stringent checking of all input values. The implementation has embedded system tests. Such formalism as exists in the HotSpot design and implementation process is largely reflected in the discipline maintained by the developer, who has consistently followed these approaches over long periods of time.

III.F.2. NARAC System

The majority of the components in the NARAC system are in maintenance mode with the core components having been operational for 4-6 years. However, new requirements often require new components or new interactions between existing components. In either case, significant new development is often required. Thus, development effort is a mix of incremental enhancement and original development. Note that new development will exist in the context of the existing system and both the design and implementation of new work must reflect this. The formalism associated with both incremental enhancement and new work is tuned to the scope and potential impact of the changes. Relatively minor changes require little formalism, while large-scale redesign of existing subsystems or major new components will be supported by text documents and design diagrams, generally using UML diagrams. Design reviews are part of this process.

Implementation makes extensive use of patterns and refactoring. The patterns include both industry-wide paradigms (e.g., Factory, Adaptor, Singleton) and local patterns that were developed to meet NARAC-specific requirements (e.g., Model Execution). Refactoring is an on-going effort to continuously improve the code base in terms of flexibility, clarity and ease of reuse without major redesign. Most work on incremental enhancement involves an element of refactoring. Integration of new or updated code into the core development environment, which is a fully functional instance of the system, is relatively continuous with changes being integrated on a daily to weekly basis.

III.G. Software Safety

III.G.1. HotSpot

HotSpot has been evaluated for inclusion in the DOE Safety Software Toolkit and has provisionally been accepted contingent on several refinements in procedures and documentation (these are described in the appropriate sections of this document) along with a new function that is being added to the code. HotSpot is commonly used for safety analyses at DOE sites and completing the critical recommendations for inclusion in the Safety Software suite will be a major milestone in confirming HotSpot role as an important tool in the DOE community.

III.G.2. NARAC System

While the NARAC system is not normally used in safety analyses, many of the SQA principles associated with safety software are utilized as a matter of course. All critical components and data paths are redundant to allow the system to continue to run and produce valid products even while experiencing component failure. This redundancy is continuously evaluated on the basis of experience and new technological options so that the system is constantly improving its effective availability.

Common safety software design techniques such as decoupling, isolation, redundancy, fault detection and self-diagnostics are used throughout the system. The most significant challenge in this area is in managing complexity. Given the existing breadth and depth of the NARAC mission and the growth in requirements for higher fidelity modeling, it is only occasionally possible to explicitly simplify an existing component. The vast majority of the functionality is required. Consequently, the key issue is to manage the growth in complexity with increasing effectiveness through increasingly mature design and implementation that reflects both deep understanding of the computer language and related tools as well as of scientific, technical and customer requirements associated with the NARAC mission.

III.H. Verification and Validation

III.H.1. HotSpot

HotSpot includes built-in testing as part the standard deployment that tests most of the key components of the model. In addition, the developer performs more detailed tests as part of development. A NARAC model developer also evaluates the model and all test results as part of the integration of HotSpot into the NARAC systems. The overall HotSpot system is also reviewed by NARAC system developers as part of its integration into the NARAC Web and iClient capabilities. This independent review of the testing process does need additional formalization, which will be achieved as part of the integration of HotSpot procedures with corresponding NARAC procedures.

III.H.2. NARAC System

Testing is as integral part of all aspects of the system. As described in the Introduction, the Model Research and Development team is highly focused on developer testing, model validation and verification and maintaining repeatable benchmarks suites of model runs. Most of the activities of Operations are daily exercises of system capabilities and constitute on-going testing. In addition, Operations staff are an integral part of the overall development cycle and particularly in approving a new production update for release.

Within the development team, seven levels of testing can be identified:

1. *Developer testing* – individual developers run custom tests that exercise new or changed behavior. The testing occurs in two phases. The first occurs in a private environment where the behavior of the core development system is not affected. Once this initial phase is completed, the new software is integrated into the core development system and evaluated in the context of the complete system.
2. *Unit testing* – many portions of the software have unit tests. These include both custom tests and tests built into the JUnit test harness. These are run episodically, particular when significant changes in the supporting packages are being integrated.
3. *Integration testing* – verifies that new or changed code works in the context of larger subsystems. These tests are often scripted and integrated with benchmark suites that check correct execution of the subsystem for both new and existing functionality.
4. *System testing* – there are two approaches to system testing. One is based on WinRunner, which supports scripted execution of GUI components that users rely on to control the system. Also, WinRunner supports controlled random selection of GUI options. This randomness is effective in traversing rarely taken paths through the system. The other approach relies on XML requests for model executions that are submitted by the NARAC Web and iClient systems to the core modeling system. A suite of these XML requests is maintained that spans the behavior of the system that is directly exposed to external customers. A tool exists that allows subsets of these requests to be submitted. These are run regularly against the

system to validate that any changes made have not caused problems for core system functionality.

5. *Load testing* – the same tool for managing XML requests also supports overlapping requests with a controlled delay between each request. This delay can be reduced to zero. This supports load testing of the system and is invaluable in ensuring the general thread-safety of the overall system and the correct behavior of the explicitly multi-threaded components.
6. *Acceptance testing* – acceptance of a new production update is the responsibility of Operations staff. Operations focuses on new and changed functionality as described in Production Updates notes that are produced from the Bugzilla requirements tracking system. These notes provide a checklist that ensures that all changed capabilities are tested. The status of the testing is reported in Bugzilla with failures being returned to the responsible developer. For the NARAC Web and iClient this beta testing phase includes deployment to selected external customers for their evaluation.
7. *System status testing* – the last tier of testing is an on-going evaluation of the system via a variety of monitoring tools as described previously. When defects do slip into Production, these tools can be very helpful in identifying and diagnosing the problem. More commonly, they are useful in identifying failures in external data sources and networks. NARAC is often the first to identify trouble spots both in LLNL maintained networks and in external data distribution systems. Any weaknesses in the system identified by the monitoring are evaluated in regular meetings as well as in hot washes after real responses and major exercises.

III.I. Problem Reporting

III.I.1 HotSpot

The HotSpot developer has managed problem reporting using e-mail. While this has been successful for most purposes, a more formal procedure is desirable. HotSpot problem reporting is being integrated into the NARAC Bugzilla system.

III.I.2. NARAC System

In addition to the customized use of Bugzilla for various requirements and testing tasks, it is also used for its original purpose as a defect tracking system. Problems are entered into Bugzilla, assigned to a developer who estimates the resources involved in correcting the problem. This effort is then prioritized with other activities primarily based on discussions between Operations and Systems Development. From this point, it is managed like other changes in the system. For problems that affect the Production systems in critical ways. The formal process can be short-circuited when Operations identifies a critical problem and contacts the responsible developer directly. The developer works with Operations to determine the lowest risk solution to the problem. The problem and resolution are entered into Bugzilla.

III.J. Training

III.J.1 HotSpot

The HotSpot web site includes significant user documentation and the code itself had significant help features. The developer is a recognized expert in the field and accessible at conferences and via e-mail. Due to the integration of HotSpot with NARAC, a growing number of NARAC staff members are familiar with HotSpot.

III.J.2. NARAC System

Internal training of NARAC Operations staff takes the form of formal presentations, personal interactions and written documentation. While not universal, a significant amount of information is available on the internal NARAC web site. Also, since most Operations activities involve use of the operational tools, these activities are an important aspect of training.

External users of the system are supported by documentation available on the NARAC Web page. The Customer Support component of the Operations Team focuses substantial effort on educating users and guides them through parts of the system they find confusing. NARAC staff travel to external sites to give training classes and typically twice a year give training classes to groups of 20-30 at NARAC.

IV. CONCLUSIONS

NARAC has a mature set of SQA procedures that are balanced with other aspects of the broader issue of quality assurance. This balance is always being evaluated to attempt to maximize the value of the overall system within the various resource constraints that exist at a given time.

ACKNOWLEDGEMENTS

Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344.

REFERENCES

1. http://narac.llnl.gov/
2. Nasstrom, J.S., G. Sugiyama, R. L. Baskett, S.C. Larsen, M.M. Bradley, 2006: The NARAC Modeling and Decision Support System for Radiological and Nuclear Emergency Preparedness and Response. *Int. J. Emergency Management*, **V. 4**, No. 3, 2006
3. DOE, U.S. Department of Energy, *Quality Assurance*, DOE O 414.1C, (2005)
4. ASME, American Society of Mechanical Engineers, *Quality Assurance Requirements for Nuclear Facilities*, NQA-1-2000.